```
#!/usr/bin/perl -w
use strict;

#FILE:    cmah.pl
#PROJECT: Reproductive Incompatibility by Immunity
#AUTHOR:  Stevan Springer

#SECTION 0 - INITIALIZATION
#0a - Subroutines
use sub_calculate_binomial_probability;
use sub_calculate_n_choose_k;

#0b - Command line input/output
my $time_start;
my $outfile;

#0c - Model parameters
my $increment_q = 0.01;
my $n = 1;
my $k = 0;
my $mating_attempts = 20;
my $compatibility = 0;

#0d - Model variables
my $q;

my $m_s;
my $m_q_nextgen;
my $m_delta_q;
my $m_percent_delta_q;

my $f_s;
my $f_q_nextgen;
my $f_delta_q;
my $f_percent_delta_q;
my $encounter_prob;
my $prob_unmated;

my $net_s;
my $net_delta_q;
my $net_percent_delta_q;
my $past_net_s;
my $past_net_delta_q = -1;

#0e - Read control file name from @ARGV, open and read into @input_control
if (scalar(@ARGV) != 0) {
       print(STDERR
"----------
Usage: cmah.pl
----------
Description: Analytical model of immune mediated incompatibility.
Input: No control file. Specify parameters by changing values directly in cmah.pl\n\n");
       exit(1);
}

#SECTION 1 - MODEL
#      Payoff Matrix: q is the frequency of the allele for the loss of sialic-acid.
#      Females have a non-self immune reaction to the sperm of (+/+) and (+/-) males,
#
#                                Female
#                         ++     +-     --
#                   ++     1      1      0
#      Male         +-     1      1      0
#                   --     1      1      1
```

```
#1a - Program execution
print "Matings ZeroPoint Compatibility = $compatibility\n";

while ($n <= $mating_attempts) {        #Calculate deltas for n mating attempts
        $outfile = "output/Compatibility=$compatibility-Matings=$n.xls";
        open(OUTPUT, "+>>$outfile") or die "can't open $outfile $!";
        print OUTPUT "q\tnet_delta_q\tm_delta_q\tf_delta_q\tq\tnet_s\tm_s\tf_s\t$time_start
                        Program: cmah.pl Mating Attempts: $n\n";

        $q = $increment_q;                                          #Start at lowest increment
        while ($q < 1) {

#1b - Calculate the dynamics for males
                $m_s = (1 - $compatibility) * $q**2;
                        #Proportion of females compatible with a (-/-) male = 1, Proportion
                        #compatible with a (+/+) or (+/-) male = 1-q^2. q^2 = s_m, the
                        #disadvantage to (+/+) and (+/-) males.
                $m_q_nextgen = ($q - ($m_s * $q) + ($m_s * $q**2)) / (1 - ($m_s * (1 - $q**2)));
                        #From Falconer, page 28, scenario 4, selection against dominant allele.
                $m_delta_q = $m_q_nextgen - $q;
                $m_percent_delta_q = $m_delta_q * 100;

#1c - Calculate the dynamics for females
                $encounter_prob = $q**2 + ($compatibility * (1-$q**2));
                        #(-/-) females are only compatible with (-/-) males, therefore the
                        #probability of encountering a (-/-) male in each mating attempt is q
                        #squared.
                $prob_unmated = calculate_binomial_probability($n, $k, $encounter_prob);
                        #Probability that a (-/-) female encounters k=0 compatible mates in n
                        #matings.
                $f_s = $prob_unmated;
                        #Proportion of (-/-) females that did not mate (-s) relative to (+/+) and
                        #(+/-) females (1). Note that the proportion that mated is 1-s therefore
                        #the proportion that are unmated is s_female.
                $f_q_nextgen = ($q - ($f_s * $q**2)) / (1 - ($f_s * $q**2));
                        #Equation 3 from page 28 of Falconer.
                $f_delta_q = $f_q_nextgen - $q;
                $f_percent_delta_q = $f_delta_q * 100;


#1d - Calculate net selection, net direction of frequency change and zero points
                $net_s = $m_s - $f_s;
                $net_delta_q = $m_delta_q + $f_delta_q;
                $net_percent_delta_q = $net_delta_q * 100;

                print OUTPUT "$q\t$net_percent_delta_q\t$m_percent_delta_q\t$f_percent_delta_q
                             \t$q\t$net_s\t$m_s\t-$f_s\n";
                if ($net_s >= 0 and $past_net_s < 0) {
                        print "ZeroPoint_s: $n $q\n";
                }
                if ($net_delta_q >= 0 and $past_net_delta_q < 0) {
#                       print "Zero Point Delta q: $n = $q\n";
                }
                $past_net_s = $net_s;
                $past_net_delta_q = $net_delta_q;

#1e - Increment $q and $n counters
                $q = sprintf("%.3f", $q + $increment_q);

        }
        $n++;
        close OUTPUT;
}
```

```perl
#FILE: sub_calculate_binomial_probability.pm
#SUBROUTINE: CALCULATE_BINOMIAL_PROBABILITY
#Input: # of attempts (n), # of occurrences (k), and probability of occurrence in each
attempt (p).
#Return: The probability of the event occuring k times.

sub calculate_binomial_probability {

my ($n,$k,$p) = @_;

#Method of calculating binomial probability
return $k == 0 if $p ==0;
return $k != $n if $p == 1;
return calculate_n_choose_k($n, $k) * $p**$k * (1-$p)**($n-$k);

}

1
```

```perl
#FILE: sub_calculate_n_choose_k.pm
#SUBROUTINE: CALCULATE_N_CHOOSE_K
#Input: size of the total set (n), # of elements to draw from the set (k).
#Return: The number of ways to choose k elements from a set of n elements.

sub calculate_n_choose_k {


#calculate_n_choose_k($n, $k) is the number of ways to choose $k elements from a #set of $n
elements, when the order of selection in irrelevant.

my ($n,$k) = @_;
my ($result, $j) = (1, 1);


return 0 if $k > $n || $k < 0;
$k = ($n - $k) if ($n - $k) < $k;

while ( $j <= $k ) {
       $result *= $n--;
       $result /= $j++;
}
return $result;

}

1
```